Phrase-based Statistical Language Generation using Graphical Models and Active Learning

François Mairesse, **Milica Gašić**, **Filip Jurčíček**,
Simon Keizer, **Blaise Thomson**, **Kai Yu** and **Steve Young**[*]
Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK
{f.mairesse, mg436, fj228, sk561, brmt2, ky219, sjy}@eng.cam.ac.uk

Abstract

Most previous work on trainable language generation has focused on two paradigms: (a) using a statistical model to rank a set of generated utterances, or (b) using statistics to inform the generation decision process. Both approaches rely on the existence of a handcrafted generator, which limits their scalability to new domains. This paper presents BAGEL, a statistical language generator which uses dynamic Bayesian networks to learn from semantically-aligned data produced by 42 untrained annotators. A human evaluation shows that BAGEL can generate natural and informative utterances from *unseen* inputs in the information presentation domain. Additionally, generation performance on sparse datasets is improved significantly by using certainty-based active learning, yielding ratings close to the human gold standard with a fraction of the data.

1 Introduction

The field of natural language generation (NLG) is one of the last areas of computational linguistics to embrace statistical methods. Over the past decade, statistical NLG has followed two lines of research. The first one, pioneered by Langkilde and Knight (1998), introduces statistics in the generation process by training a model which reranks candidate outputs of a handcrafted generator. While their HALOGEN system uses an n-gram language model trained on news articles, other systems have used hierarchical syntactic models (Bangalore and Rambow, 2000), models trained on user ratings of utterance quality (Walker et al., 2002), or alignment models trained on speaker-specific corpora (Isard et al., 2006).

A second line of research has focused on introducing statistics at the generation decision level, by training models that find the set of generation parameters maximising an objective function, e.g. producing a target linguistic style (Paiva and Evans, 2005; Mairesse and Walker, 2008), generating the most likely context-free derivations given a corpus (Belz, 2008), or maximising the expected reward using reinforcement learning (Rieser and Lemon, 2009). While such methods do not suffer from the computational cost of an overgeneration phase, they still require a handcrafted generator to define the generation decision space within which statistics can be used to find an optimal solution.

This paper presents BAGEL (Bayesian networks for generation using active learning), an NLG system that can be fully trained from aligned data. While the main requirement of the generator is to produce natural utterances within a dialogue system domain, a second objective is to minimise the overall development effort. In this regard, a major advantage of data-driven methods is the shift of the effort from model design and implementation to data annotation. In the case of NLG systems, learning to produce paraphrases can be facilitated by collecting data from a large sample of annotators. Our meaning representation should therefore (a) be intuitive enough to be understood by untrained annotators, and (b) provide useful generalisation properties for generating unseen inputs. Section 2 describes BAGEL's meaning representation, which satisfies both requirements. Section 3 then details how our meaning representation is mapped to a phrase sequence, using a dynamic Bayesian network with backoff smoothing.

Within a given domain, the same semantic concept can occur in different utterances. Section 4 details how BAGEL exploits this redundancy

[*] This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and funded by the EU FP7 Programme under grant agreement 216594 (CLASSiC project: www.classic-project.org).

to improve generation performance on sparse datasets, by guiding the data collection process using *certainty-based active learning* (Lewis and Catlett, 1994). We train BAGEL in the information presentation domain, from a corpus of utterances produced by 42 untrained annotators (see Section 5.1). An automated evaluation metric is used to compare preliminary model and training configurations in Section 5.2, while Section 5.3 shows that the resulting system produces natural and informative utterances, according to 18 human judges. Finally, our human evaluation shows that training using active learning significantly improves generation performance on sparse datasets, yielding results close to the human gold standard using a fraction of the data.

2 Phrase-based generation from semantic stacks

BAGEL uses a *stack-based* semantic representation to constrain the sequence of semantic concepts to be searched. This representation can be seen as a linearised semantic tree similar to the one previously used for natural language understanding in the Hidden Vector State model (He and Young, 2005). A stack representation provides useful generalisation properties (see Section 3.1), while the resulting stack sequences are relatively easy to align (see Section 5.1). In the context of dialogue systems, Table 1 illustrates how the input dialogue act is first mapped to a set of stacks of semantic concepts, and then aligned with a word sequence. The bottom concept in the stack will typically be a *dialogue act type*, e.g. an utterance providing information about the object under discussion (`inform`) or specifying that the request of the user cannot be met (`reject`). Other concepts include attributes of that object (e.g., `food`, `area`), values for those attributes (e.g., `Chinese`, `riverside`), as well as special symbols for negating underlying concepts (e.g., `not`) or specifying that they are irrelevant (e.g., `dontcare`).

The generator's goal is thus finding the most likely realisation given an *unordered* set of *mandatory* semantic stacks $\mathcal{S}_m$ derived from the input dialogue act. For example, $s =$`inform(area(centre))` is a mandatory stack associated with the dialogue act in Table 1 (frame 8). While mandatory stacks must all be conveyed in the output realisation, $\mathcal{S}_m$ does not contain the optional *intermediary* stacks $\mathcal{S}_i$ that can refer to (a) general attributes of the object under discussion (e.g., `inform(area)` in Table 1), or (b) to concepts that are not in the input at all, which are associated with the singleton stack `inform` (e.g., phrases expressing the dialogue act type, or clause aggregation operations). For example, the stack sequence in Table 1 contains 3 intermediary stacks for $t = 2$, 5 and 7.

BAGEL's granularity is defined by the semantic annotation in the training data, rather than external linguistic knowledge about what constitutes a unit of meaning, i.e. contiguous words belonging to the same semantic stack are modelled as an atomic observation unit or *phrase*.[1] In contrast with word-level models, a major advantage of phrase-based generation models is that they can model long-range dependencies and domain-specific idiomatic phrases with fewer parameters.

3 Dynamic Bayesian networks for NLG

Dynamic Bayesian networks have been used successfully for speech recognition, natural language understanding, dialogue management and text-to-speech synthesis (Rabiner, 1989; He and Young, 2005; Lefèvre, 2006; Thomson and Young, 2010; Tokuda et al., 2000). Such models provide a principled framework for predicting elements in a large structured space, such as required for non-trivial NLG tasks. Additionally, their probabilistic nature makes them suitable for modelling linguistic variation, i.e. there can be multiple valid paraphrases for a given input.

BAGEL models the generation task as finding the most likely sequence of realisation phrases $\mathbf{R}^* = (r_1...r_L)$ given an unordered set of mandatory semantic stacks $\mathcal{S}_m$, with $|\mathcal{S}_m| \leq L$. BAGEL must thus derive the optimal sequence of semantic stacks $\mathbf{S}^*$ that will appear in the utterance given $\mathcal{S}_m$, i.e. by inserting intermediary stacks if needed and by performing content ordering. Any number of intermediary stacks can be inserted between two consecutive mandatory stacks, as long as all their concepts are included in either the previous or following mandatory stack, and as long as each stack transition leads to a different stack (see example in Table 1). Let us define the set of possible stack sequences matching these constraints as $Seq(\mathcal{S}_m) \subseteq \{\mathbf{S} = (s_1...s_L) \text{ s.t. } s_t \in \mathcal{S}_m \cup \mathcal{S}_i\}$.

We propose a model which estimates the dis-

[1] The term *phrase* is thus defined here as any sequence of one or more words.

Charlie Chan	*is a*	*Chinese*	*restaurant*	*near*	*Cineworld*	*in the*	*centre of town*
Charlie Chan **name** **inform**	inform	**Chinese** **food** **inform**	**restaurant** **type** **inform**	near inform	**Cineworld** **near** **inform**	area inform	**centre** **area** **inform**
$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$

Table 1: Example semantic stacks aligned with an utterance for the dialogue act `inform(name(Charlie Chan) type(restaurant) area(centre) food(Chinese) near(Cineworld))`. Mandatory stacks are in bold.

tribution $P(\mathbf{R}|\mathcal{S}_m)$ from a training set of realisation phrases aligned with semantic stack sequences, by marginalising over all stack sequences in $Seq(\mathcal{S}_m)$:

$$\begin{aligned} P(\mathbf{R}|\mathcal{S}_m) &= \sum_{\mathbf{S}\in Seq(\mathcal{S}_m)} P(\mathbf{R},\mathbf{S}|\mathcal{S}_m) \\ &= \sum_{\mathbf{S}\in Seq(\mathcal{S}_m)} P(\mathbf{R}|\mathbf{S},\mathcal{S}_m)P(\mathbf{S}|\mathcal{S}_m) \\ &= \sum_{\mathbf{S}\in Seq(\mathcal{S}_m)} P(\mathbf{R}|\mathbf{S})P(\mathbf{S}|\mathcal{S}_m) \qquad (1) \end{aligned}$$

Inference over the model defined in (1) requires the decoding algorithm to consider all possible orderings over $Seq(\mathcal{S}_m)$ together with all possible realisations, which is intractable for non-trivial domains. We thus make the additional assumption that the most likely sequence of semantic stacks $\mathbf{S}^*$ given $\mathcal{S}_m$ is the one yielding the optimal realisation phrase sequence:

$$\begin{aligned} P(\mathbf{R}|\mathcal{S}_m) &\approx P(\mathbf{R}|\mathbf{S}^*)P(\mathbf{S}^*|\mathcal{S}_m) \qquad (2) \\ \text{with } \mathbf{S}^* &= \operatorname*{argmax}_{\mathbf{S}\in Seq(\mathcal{S}_m)} P(\mathbf{S}|\mathcal{S}_m) \qquad (3) \end{aligned}$$

The semantic stacks are therefore decoded first using the model in Fig. 1 to solve the argmax in (3). The decoded stack sequence $\mathbf{S}^*$ is then treated as observed in the realisation phase, in which the model in Fig. 2 is used to find the realisation phrase sequence $\mathbf{R}^*$ maximising $P(\mathbf{R}|\mathbf{S}^*)$ over all phrase sequences of length $L = |\mathbf{S}^*|$ in our vocabulary:

$$\begin{aligned} \mathbf{R}^* &= \operatorname*{argmax}_{\mathbf{R}=(r_1\ldots r_L)} P(\mathbf{R}|\mathbf{S}^*)P(\mathbf{S}^*|\mathcal{S}_m) \qquad (4) \\ &= \operatorname*{argmax}_{\mathbf{R}=(r_1\ldots r_L)} P(\mathbf{R}|\mathbf{S}^*) \qquad (5) \end{aligned}$$

In order to reduce model complexity, we factorise our model by conditioning the realisation phrase at time t on the previous phrase r_{t-1}, and the previous, current, and following semantic stacks. The semantic stack s_t at time t is assumed to depend only on the previous two stacks and the last mandatory stack $s_u \in \mathcal{S}_m$ with $1 \leq u < t$:

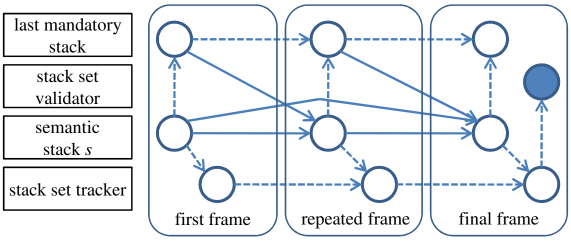

Figure 1: Graphical model for the semantic decoding phase. Plain arrows indicate smoothed probability distributions, dashed arrows indicate deterministic relations, and shaded nodes are observed. The generation of the `end` semantic stack symbol deterministically triggers the final frame.

$$P(\mathbf{S}|\mathcal{S}_m) = \begin{cases} \prod_{t=1}^{T} P(s_t|s_{t-1}, s_{t-2}, s_u) \\ \quad \text{if } \mathbf{S} \in Seq(\mathcal{S}_m) \\ 0 \quad \text{otherwise} \end{cases} \qquad (6)$$

$$P(\mathbf{R}|\mathbf{S}^*) = \prod_{t=1}^{T} P(r_t|r_{t-1}, s^*_{t-1}, s^*_t, s^*_{t+1}) \qquad (7)$$

While dynamic Bayesian networks typically take sequential inputs, mapping a *set* of semantic stacks to a sequence of phrases is achieved by keeping track of the mandatory stacks that were visited in the current sequence (see stack set tracker variable in Fig. 1), and pruning any sequence that has not included all mandatory input stacks on reaching the final frame (see observed stack set validator variable in Fig. 1). Since the number of intermediary stacks is not known at decoding time, the network is unrolled for a fixed number of frames T defining the maximum number of phrases that can be generated (e.g., $T = 50$). The end of the stack sequence is then determined by a special `end` symbol, which can only be emitted within the T frames once all mandatory stacks have been visited. The probability of the resulting utterance is thus computed over all frames up to the `end` symbol, which determines the length

L of $\mathbf{S}^*$ and $\mathbf{R}^*$. While the decoding constraints enforce that $L > |\mathcal{S}_m|$, the search for $\mathbf{S}^*$ requires comparing sequences of different lengths. A consequence is that shorter sequences containing only mandatory stacks are likely to be favoured. While future work should investigate length normalisation strategies, we find that the learned transition probabilities are skewed enough to favour stack sequences including intermediary stacks.

Once the topology and the decoding constraints of the network have been defined, any inference algorithm can be used to search for $\mathbf{S}^*$ and $\mathbf{R}^*$. We use the junction tree algorithm implemented in the Graphical Model ToolKit (GMTK) for our experiments (Bilmes and Zweig, 2002), however both problems can be solved using a standard Viterbi search given the appropriate state representation. In terms of computational complexity, it is important to note that the number of stack sequences $Seq(\mathcal{S}_m)$ to search over increases exponentially with the number of input mandatory stacks. Nevertheless, we find that real-time performance can be achieved by pruning low probability sequences, without affecting the quality of the solution.

3.1 Generalisation to unseen semantic stacks

In order to generalise to semantic stacks which have not been observed during training, the realisation phrase r is made dependent on underspecified stack configurations, i.e. the tail l and the head h of the stack. For example, the last stack in Table 1 is associated with the head `centre` and the tail `inform(area)`. As a result, BAGEL assigns non-zero probabilities to realisation phrases in unseen semantic contexts, by backing off to the head and the tail of the stack. A consequence is that BAGEL's lexical realisation can generalise across contexts. For example, if `reject(area(centre))` was never observed at training time, $P(r = \text{centre of town}|s = \texttt{reject(area(centre))})$ will be estimated by backing off to $P(r = \text{centre of town}|h = \texttt{centre})$. BAGEL can thus generate 'there are no venues in the centre of town' if the phrase 'centre of town' was associated with the concept `centre` in a different context, such as `inform(area(centre))`. The final realisation model is illustrated in Fig. 2:

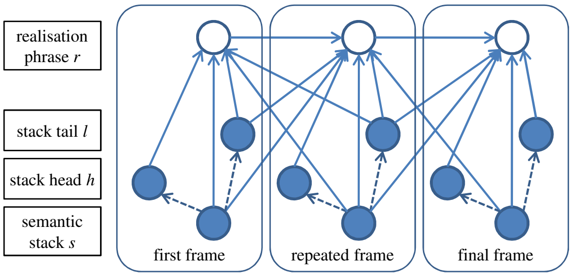

Figure 2: Graphical model for the realisation phase. Dashed arrows indicate deterministic relations, and shaded node are observed.

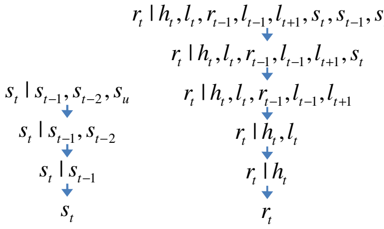
$s_t \mid s_{t-1}, s_{t-2}, s_u$
$s_t \mid s_{t-1}, s_{t-2}$
$s_t \mid s_{t-1}$
s_t
$r_t \mid h_t, l_t, r_{t-1}, l_{t-1}, l_{t+1}, s_t, s_{t-1}, s_{t+1}$
$r_t \mid h_t, l_t, r_{t-1}, l_{t-1}, l_{t+1}, s_t$
$r_t \mid h_t, l_t, r_{t-1}, l_{t-1}, l_{t+1}$
$r_t \mid h_t, l_t$
$r_t \mid h_t$
r_t

Figure 3: Backoff graphs for the semantic decoding and realisation models.

$$P(\mathbf{R}|\mathbf{S}^*) = \prod_{t=1}^{L} P(r_t|r_{t-1}, h_t, l_{t-1}, l_t, l_{t+1}, s^*_{t-1}, s^*_t, s^*_{t+1}) \quad (8)$$

Conditional probability distributions are represented as factored language models smoothed using Witten-Bell interpolated backoff smoothing (Bilmes and Kirchhoff, 2003), according to the backoff graphs in Fig. 3. Variables which are the furthest away in time are dropped first, and partial stack variables are dropped last as they are observed the most.

It is important to note that generating unseen semantic stacks requires all possible mandatory semantic stacks in the target domain to be predefined, in order for all stack unigrams to be assigned a smoothed non-zero probability.

3.2 High cardinality concept abstraction

While one should expect a trainable generator to learn multiple lexical realisations for low-cardinality semantic concepts, learning lexical realisations for high-cardinality database entries (e.g., proper names) would increase the number of model parameters prohibitively. We thus divide pre-terminal concepts in the semantic stacks into two types: (a) *enumerable* attributes whose values are associated with distinct semantic stacks in our model (e.g., `inform(pricerange(cheap))`), and (b) *non-enumerable* attributes whose values are replaced by a generic symbol before training in both the utterance and the semantic stack (e.g., `inform(name(X))`. These symbolic values are then replaced in the surface realisation by the corresponding value in the input specification. A consequence is that our model can only learn synonymous lexical realisations for enumerable attributes.

4 Certainty-based active learning

A major issue with trainable NLG systems is the lack of availability of domain-specific data. It is therefore essential to produce NLG models that minimise the data annotation cost.

BAGEL supports the optimisation of the data collection process through active learning, in which the next semantic input to annotate is determined by the current model. The probabilistic nature of BAGEL allows the use of *certainty-based* active learning (Lewis and Catlett, 1994), by querying the k semantic inputs for which the model is the least certain about its output realisation. Given a finite semantic input space $\mathcal{I}$ representing all possible dialogue acts in our domain (i.e., the set of all sets of mandatory semantic stacks $\mathcal{S}_m$), BAGEL's active learning training process iterates over the following steps:

1. Generate an utterance for each semantic input $\mathcal{S}_m \in \mathcal{I}$ using the current model.[2]

2. Annotate the k semantic inputs $\{\mathcal{S}_m^1 ... \mathcal{S}_m^k\}$ yielding the lowest realisation probability, i.e. for $q \in (1..k)$

$$\mathcal{S}_m^q = \underset{\mathcal{S}_m \in \mathcal{I} \setminus \{\mathcal{S}_m^1 ... \mathcal{S}_m^{q-1}\}}{\operatorname{argmin}} (\max_{\mathbf{R}} P(\mathbf{R}|\mathcal{S}_m)) \qquad (9)$$

   with $P(\mathbf{R}|\mathcal{S}_m)$ defined in (2).

3. Retrain the model with the additional k data points.

The number of utterances to be queried k should depend on the flexibility of the annotators and the time required for generating all possible utterances in the domain.

5 Experimental method

BAGEL's factored language models are trained using the SRILM toolkit (Stolcke, 2002), and decoding is performed using GMTK's junction tree inference algorithm (Bilmes and Zweig, 2002). Since each active learning iteration requires generating all training utterances in our domain, they are generated using a larger clique pruning threshold than the test utterances used for evaluation.

5.1 Corpus collection

We train BAGEL in the context of a dialogue system providing information about restaurants in Cambridge. The domain contains two dialogue act types: (a) `inform`: presenting information about a restaurant (see Table 1), and (b) `reject`: informing that the user's constraints cannot be met (e.g., 'There is no cheap restaurant in the centre'). Our domain contains 8 restaurant attributes: `name`, `food`, `near`, `pricerange`, `postcode`, `phone`, `address`, and `area`, out of which `food`, `pricerange`, and `area` are treated as enumerable.[3] Our input semantic space is approximated by the set of information presentation dialogue acts produced over 20,000 simulated dialogues between our statistical dialogue manager (Young et al., 2010) and an agenda-based user simulator (Schatzmann et al., 2007), which results in 202 unique dialogue acts after replacing non-enumerable values by a generic symbol. Each dialogue act contains an average of 4.48 mandatory semantic stacks.

As one of our objectives is to test whether BAGEL can learn from data provided by a large sample of untrained annotators, we collected a corpus of semantically-aligned utterances using Amazon's Mechanical Turk data collection service. A crucial aspect of data collection for NLG is to ensure that the annotators understand the meaning of the semantics to be conveyed. Annotators were first asked to provide an utterance matching an abstract description of the dialogue act, regardless of the order in which the constraints are presented (e.g., *Offer the venue Taj Mahal and provide the information type(restaurant), area(riverside), food(Indian), near(The Red Lion)*). The order of the constraints in the description was randomised to reduce the effect of priming. The annotators were then asked to align the attributes (e.g., *Indicate the region of the utterance related to the concept 'area'*), and the attribute values (e.g., *Indicate only the words related to the concept 'riverside'*). Two paraphrases were collected for each dialogue act in our domain, resulting in a total of 404 aligned ut-

[2] Sampling methods can be used if $\mathcal{I}$ is infinite or too large.

[3] With the exception of areas defined as proper nouns.

r_t	s_t	h_t	l_t
<s>	START	START	START
The Rice Boat	inform(name(X))	X	inform(name)
is a	inform	inform	EMPTY
restaurant	inform(type(restaurant))	restaurant	inform(type)
in the	inform(area)	area	inform
riverside	inform(area(riverside))	riverside	inform(area)
area	inform(area)	area	inform
that	inform	inform	EMPTY
serves	inform(food)	food	inform
French	inform(food(French))	French	inform(food)
food	inform(food)	food	inform
</s>	END	END	END

Table 2: Example utterance annotation used to estimate the conditional probability distributions of the models in Figs. 1 and 2 (r_t=realisation phrase, s_t=semantic stack, h_t=stack head, l_t=stack tail).

terances produced by 42 native speakers of English. After manually checking and normalising the dataset,[4] the layered annotations were automatically mapped to phrase-level semantic stacks by splitting the utterance into phrases at annotation boundaries. Each annotated utterance is then converted into a sequence of symbols such as in Table 2, which are used to estimate the conditional probability distributions defined in (6) and (8). The resulting vocabulary consists of 52 distinct semantic stacks and 109 distinct realisation phrases, with an average of 8.35 phrases per utterance.

5.2 BLEU score evaluation

We first evaluate BAGEL using the BLEU automated metric (Papineni et al., 2002), which measures the word n-gram overlap between the generated utterances and the 2 reference paraphrases over a test corpus (with n up to 4). While BLEU suffers from known issues such as a bias towards statistical NLG systems (Reiter and Belz, 2009), it provides useful information when comparing similar systems. We evaluate BAGEL for different training set sizes, model dependencies, and active learning parameters. Our results are averaged over a 10-fold cross-validation over distinct dialogue acts, i.e. dialogue acts used for testing are *not seen* at training time,[5] and all systems are tested on the same folds. The training and test sets respectively contain an average of 181 and 21 distinct dialogue acts, and each dialogue act is associated with two paraphrases, resulting in 362 training utterances.

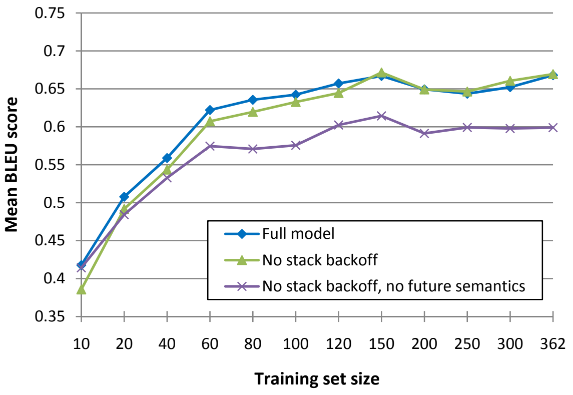

Figure 4: BLEU score averaged over a 10-fold cross-validation for different training set sizes and network topologies, using random sampling.

Results: Fig. 4 shows that adding a dependency on the future semantic stack improves performances for all training set sizes, despite the added model complexity. Backing off to partial stacks also improves performance, but only for sparse training sets.

Fig. 5 compares the full model trained using random sampling in Fig. 4 with the same model trained using certainty-based active learning, for different values of k. As our dataset only contains two paraphrases per dialogue act, the same dialogue act can only be queried twice during the active learning procedure. A consequence is that the training set used for active learning converges towards the randomly sampled set as its size increases. Results show that increasing the training set one utterance at a time using active learning ($k = 1$) significantly outperforms random sampling when using 40, 80, and 100 utterances ($p < .05$, two-tailed). Increasing the number of utterances to be queried at each iteration to $k = 10$ results in a smaller performance increase. A possi-

[4] The normalisation process took around 4 person-hour for 404 utterances.

[5] We do not evaluate performance on dialogue acts used for training, as the training examples can trivially be used as generation templates.

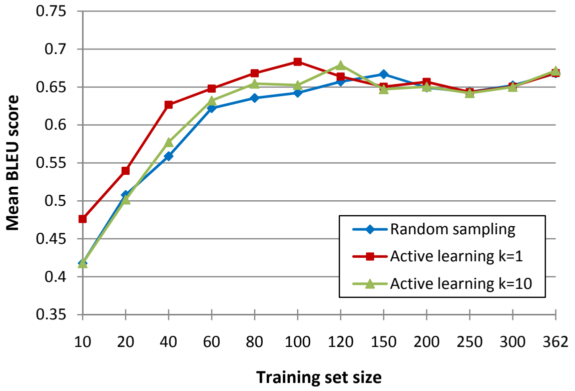

Figure 5: BLEU score averaged over a 10-fold cross-validation for different numbers of queries per iteration, using the full model with the query selection criterion (9).

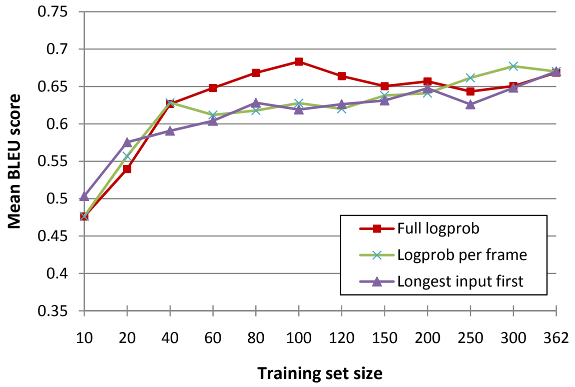

Figure 6: BLEU score averaged over a 10-fold cross-validation for different query selection criteria, using the full model with $k = 1$.

ble explanation is that the model is likely to assign low probabilities to similar inputs, thus any value above $k = 1$ might result in redundant queries within an iteration.

As the length of the semantic stack sequence is not known before decoding, the active learning selection criterion presented in (9) is biased towards longer utterances, which tend to have a lower probability. However, Fig. 6 shows that normalising the log probability by the number of semantic stacks does not improve overall learning performance. Although a possible explanation is that longer inputs tend to contain more information to learn from, Fig. 6 shows that a baseline selecting the largest remaining semantic input at each iteration performs worse than the active learning scheme for training sets above 20 utterances. The full log probability selection criterion defined in (9) is therefore used throughout the rest of the paper (with $k = 1$).

5.3 Human evaluation

While automated metrics provide useful information for comparing different systems, human feedback is needed to assess (a) the quality of BAGEL's outputs, and (b) whether training models using active learning has a significant impact on user perceptions. We evaluate BAGEL through a large-scale subjective rating experiment using Amazon's Mechanical Turk service.

For each dialogue act in our domain, participants are presented with a 'gold standard' human utterance from our dataset, which they must compare with utterances generated by models trained with and without active learning on a set of 20, 40, 100, and 362 utterances (full training set), as well as with the second human utterance in our dataset. See example utterances in Table 3. The judges are then asked to evaluate the *informativeness* and *naturalness* of each of the 8 utterances on a 5 point likert-scale. Naturalness is defined as whether the utterance could have been produced by a human, and informativeness is defined as whether it contains all the information in the gold standard utterance. Each utterance is taken from the test folds of the cross-validation experiment presented in Section 5.2, i.e. the models are trained on up to 90% of the data and the training set does not contain the dialogue act being tested.

Results: Figs. 7 and 8 compare the naturalness and informativeness scores of each system averaged over all 202 dialogue acts. A paired t-test shows that models trained on 40 utterances or less produce utterances that are rated significantly lower than human utterances for both naturalness and informativeness ($p < .05$, two-tailed). However, models trained on 100 utterances or more do not perform significantly worse than human utterances for both dimensions, with a mean difference below .10 over 202 comparisons. Given the large sample size, this result suggests that BAGEL can successfully learn our domain using a fraction of our initial dataset.

As far as the learning method is concerned, a paired t-test shows that models trained on 20 and 40 utterances using active learning significantly outperform models trained using random sampling, for both dimensions ($p < .05$). The largest increase is observed using 20 utterances, i.e. the naturalness increases by .49 and the informativeness by .37. When training on 100 utterances, the effect of active learning becomes insignificant. In-

Input	`inform(name(the Fountain) near(the Arts Picture House) area(centre) pricerange(cheap))`
Human	There is an inexpensive restaurant called the Fountain in the centre of town near the Arts Picture House
Rand-20	The Fountain is a restaurant near the Arts Picture House located in the city centre cheap price range
Rand-40	The Fountain is a restaurant in the cheap city centre area near the Arts Picture House
AL-20	The Fountain is a restaurant near the Arts Picture House in the city centre cheap
AL-40	The Fountain is an affordable restaurant near the Arts Picture House in the city centre
Full set	The Fountain is a cheap restaurant in the city centre near the Arts Picture House
Input	`reject(area(Barnwell) near(Saint Mary′s Church))`
Human	I am sorry but I know of no venues near Saint Mary's Church in the Barnwell area
Full set	I am sorry but there are no venues near Saint Mary's Church in the Barnwell area
Input	`inform(name(the Swan)area(Castle Hill) pricerange(expensive))`
Human	The Swan is a restaurant in Castle Hill if you are seeking something expensive
Full set	The Swan is an expensive restaurant in the Castle Hill area
Input	`inform(name(Browns) area(centre) near(the Crowne Plaza) near(El Shaddai) pricerange(cheap))`
Human	Browns is an affordable restaurant located near the Crowne Plaza and El Shaddai in the centre of the city
Full set	Browns is a cheap restaurant in the city centre near the Crowne Plaza and El Shaddai

Table 3: Example utterances for different input dialogue acts and system configurations. *AL-20* = active learning with 20 utterances, *Rand* = random sampling.

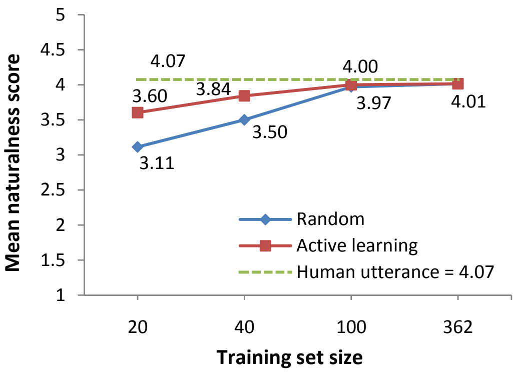

Figure 7: Naturalness mean opinion scores for different training set sizes, using random sampling and active learning. Differences for training set sizes of 20 and 40 are all significant ($p < .05$).

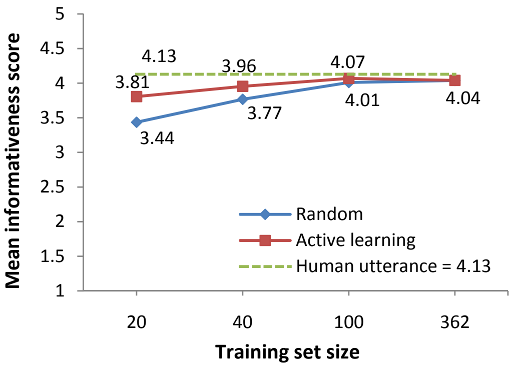

Figure 8: Informativeness mean opinion scores for different training set sizes, using random sampling and active learning. Differences for training set sizes of 20 and 40 are all significant ($p < .05$).

terestingly, while models trained on 100 utterances outperform models trained on 40 utterances using random sampling ($p < .05$), they do not significantly outperform models trained on 40 utterances using active learning ($p = .15$ for naturalness and $p = .41$ for informativeness). These results suggest that certainty-based active learning is beneficial for training a generator from a limited amount of data given the domain size.

Looking back at the results presented in Section 5.2, we find that the BLEU score correlates with a Pearson correlation coefficient of .42 with the mean naturalness score and .35 with the mean informativeness score, over all folds of all systems tested ($n = 70$, $p < .01$). This is lower than previous correlations reported by Reiter and Belz (2009) in the shipping forecast domain with non-expert judges ($r = .80$), possibly because our domain is larger and more open to subjectivity.

6 Related work

While most previous work on trainable NLG relies on a handcrafted component (see Section 1), recent research has started exploring fully data-driven NLG models.

Factored language models have recently been used for surface realisation within the OpenCCG framework (White et al., 2007; Espinosa et al., 2008). More generally, chart generators for different grammatical formalisms have been trained from syntactic treebanks (White et al., 2007; Nakanishi et al., 2005), as well as from semantically-annotated treebanks (Varges and Mellish, 2001). However, a major difference with our approach is that BAGEL uses domain-specific data to generate a surface form directly from semantic concepts, without any syntactic annotation (see Section 7 for further discussion).

This work is strongly related to Wong and Mooney's WASP^{-1} generation system (2007), which combines a language model with an inverted synchronous CFG parsing model, effectively casting the generation task as a translation problem from a meaning representation to natural language. WASP^{-1} relies on GIZA++ to align utterances with derivations of the meaning representation (Och and Ney, 2003). Although early experiments showed that GIZA++ did not perform well on our data—possibly because of the coarse granularity of our semantic representation—future work should evaluate the generalisation performance of synchronous CFGs in a dialogue system domain.

Although we do not know of any work on active learning for NLG, previous work has used active learning for semantic parsing and information extraction (Thompson et al., 1999; Tang et al., 2002), spoken language understanding (Tur et al., 2003), speech recognition (Hakkani-Tür et al., 2002), word alignment (Sassano, 2002), and more recently for statistical machine translation (Bloodgood and Callison-Burch, 2010). While certainty-based methods have been widely used, future work should investigate the performance of *committee-based* active learning for NLG, in which examples are selected based on the level of disagreement between models trained on subsets of the data (Freund et al., 1997).

7 Discussion and conclusion

This paper presents and evaluates BAGEL, a statistical language generator that can be trained entirely from data, with no handcrafting required beyond the semantic annotation. All the required subtasks—i.e. content ordering, aggregation, lexical selection and realisation—are learned from data using a unified model. To train BAGEL in a dialogue system domain, we propose a stack-based semantic representation at the phrase level, which is expressive enough to generate natural utterances from *unseen* inputs, yet simple enough for data to be collected from 42 untrained annotators with a minimal normalisation step. A human evaluation over 202 dialogue acts does not show any difference in naturalness and informativeness between BAGEL's outputs and human utterances. Additionally, we find that the data collection process can be optimised using active learning, resulting in a significant increase in performance when training data is limited, according to ratings from 18 human judges.[6] These results suggest that the proposed framework can largely reduce the development time of NLG systems.

While this paper only evaluates the most likely realisation given a dialogue act, we believe that BAGEL's probabilistic nature and generalisation capabilities are well suited to model the linguistic variation resulting from the diversity of annotators. Our first objective is thus to evaluate the quality of BAGEL's n-best outputs, and test whether sampling from the output distribution can improve naturalness and user satisfaction within a dialogue.

Our results suggest that explicitly modelling syntax is not necessary for our domain, possibly because of the lack of syntactic complexity compared with formal written language. Nevertheless, future work should investigate whether syntactic information can improve performance in more complex domains. For example, the realisation phrase can easily be conditioned on syntactic constructs governing that phrase, and the recursive nature of syntax can be modelled by keeping track of the depth of the current embedded clause. While syntactic information can be included with no human effort by using syntactic parsers, their robustness to dialogue system utterances must first be evaluated.

Finally, recent years have seen HMM-based synthesis models become competitive with unit selection methods (Tokuda et al., 2000). Our long term objective is to take advantage of those advances to jointly optimise the language generation and the speech synthesis process, by combining both components into a unified probabilistic concept-to-speech generation model.

References

S. Bangalore and O. Rambow. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 42–48, 2000.

A. Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, 2008.

J. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of HLT-NAACL, short papers*, 2003.

J. Bilmes and G. Zweig. The Graphical Models ToolKit: An open source software system for speech and time-series processing. In *Proceedings of ICASSP*, 2002.

[6] The full training corpus and the generated utterances used for evaluation are available at `http://mi.eng.cam.ac.uk/~farm2/bagel`.

M. Bloodgood and C. Callison-Burch. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.

D. Espinosa, M. White, and D. Mehay. Hypertagging: Supertagging for surface realization with CCG. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

Y. Freund, H. S. Seung, E.Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

D. Hakkani-Tür, G. Riccardi, and A. Gorin. Active learning for automatic speech recognition. In *Proceedings of ICASSP*, 2002.

Y. He and S. Young. Semantic processing using the Hidden Vector State model. *Computer Speech & Language*, 19 (1):85–106, 2005.

A. Isard, C. Brockmann, and J. Oberlander. Individuality and alignment in generated dialogues. In *Proceedings of the 4th International Natural Language Generation Conference (INLG)*, pages 22–29, 2006.

I. Langkilde and K. Knight. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 704–710, 1998.

F. Lefèvre. A DBN-based multi-level stochastic spoken language understanding system. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, 2006.

D. D. Lewis and J. Catlett. Heterogeneous uncertainty ampling for supervised learning. In *Proceedings of ICML*, 1994.

F. Mairesse and M. A. Walker. Trainable generation of Big-Five personality styles through data-driven parameter estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

H. Nakanishi, Y. Miyao, , and J. Tsujii. Probabilistic methods for disambiguation of an HPSG-based chart generator. In *Proceedings of the IWPT*, 2005.

F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

D. S. Paiva and R. Evans. Empirically-based control of natural language generation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 58–65, 2005.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

L. R. Rabiner. Tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

E. Reiter and A. Belz. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 25: 529–558, 2009.

V. Rieser and O. Lemon. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Proceedings of the Annual Meeting of the European Chapter of the ACL (EACL)*, 2009.

M. Sassano. An empirical study of active learning with support vector machines for japanese word segmentation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proceedings of HLT-NAACL, short papers*, pages 149–152, 2007.

A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, 2002.

M. Tang, X. Luo, and S. Roukos. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

C. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of ICML*, 1999.

B. Thomson and S. Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

Y. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of ICASSP*, 2000.

G. Tur, R. E. Schapire, and D. Hakkani-Tür. Active learning for spoken language understanding. In *Proceedings of ICASSP*, 2003.

S. Varges and C. Mellish. Instance-based natural language generation. In *Proceedings of the Annual Meeting of the North American Chapter of the ACL (NAACL)*, 2001.

M. A. Walker, O. Rambow, and M. Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language*, 16(3-4), 2002.

M. White, R. Rajkumar, and S. Martin. Towards broad coverage surface realization with CCG. In *Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation*, 2007.

Y. W. Wong and R. Mooney. Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of HLT-NAACL*, 2007.

S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The Hidden Information State model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, 2010.